Задача A. НОД Фибоначчи

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Числа Фибоначчи — это последовательность чисел $F(n)$, которая задается формулой: $F(0) = 1$, $F(1) = 1$, $F(n) = F(n-1) + F(n-2)$.

Даны числа a и b. Посчитайте $gcd(F(a), F(b))$ по модулю $10^9 + 7$.

Формат входных данных

Во входном файле даны неотрицательные числа a и b ($0 \leqslant a, b \leqslant 10^6$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 7	3
24 29	5

Задача B. Не быстрое не преобразование не Фурье

Имя входного файла:	`notfft.in`
Имя выходного файла:	`notfft.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Рассмотрим простое число $P = 4\,066\,273 = 2016 \cdot 2017 + 1$. Все дальнейшие вычисления в этой задаче будет проводить по модулю P. Число $g = 5$ — первообразный корень по модулю P.

Рассмотрим многочлен $A(x) = \sum_{i=0}^{n-1} a_i x^i$. Посчитаем значения многочлена в точках $1, g, g^2, \ldots, g^{n-1}$. Вам нужно найти сумму всех этих чисел.

Формат входных данных

В первой строке задано число n ($1 \leqslant n \leqslant 200\,000$). Во второй строке задано n целых чисел a_i — коэффициенты многочлена ($0 \leqslant a_i < P$).

Формат выходных данных

Выведите единственное число — $\sum_{k=0}^{n-1} A(g^k)$.

Примеры

notfft.in	notfft.out
2 1 2	14

Задача C. RSA. Взлом RSA

Имя входного файла:	`rsa.in`
Имя выходного файла:	`rsa.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В 1977 году Ronald Linn Rivest, Adi Shamir и Leonard Adleman предложили новую криптографическую схему RSA, используемую до сих пор. RSA является криптосистемой с открытым ключом: зашифровать сообщение может кто угодно, знающий общеизвестный открытый ключ, а расшифровать сообщение — только тот, кто знает специальный секретный ключ.

Желающий использовать систему RSA для получения сообщений должен сгенерировать два простых числа p и q, вычислить $n = pq$ и сгенерировать два числа e и d такие, что $ed \equiv 1 \pmod{(p-1)(q-1)}$ (заметим, что $(p-1)(q-1) = \varphi(n)$). Числа n и e составляют открытый ключ и являются общеизвестными. Число d является секретным ключом, также необходимо хранить в тайне и разложение числа n на простые множители, так как это позволяет вычислить секретный ключ d.

Сообщениями в системе RSA являются числа из $\mathbb{Z}_n$. Пусть M — исходное сообщение. Для его шифрования вычисляется значение $C = M^e \bmod n$ (для этого необходимо только знание открытого ключа). Полученное зашифрованное сообщение C передается по каналу связи. Для его расшифровки необходимо вычислить значение $M = C^d \bmod n$, а для этого необходимо знание секретного ключа.

Вы перехватили зашифрованное сообщение C и знаете только открытый ключ: числа n и e. "Взломайте" RSA — расшифруйте сообщение на основе только этих данных.

Формат входных данных

Программа получает на вход три натуральных числа: n, e, C, $n \leqslant 10^9$, $e \leqslant 10^9$, $C < n$. Числа n и e являются частью какой-то реальной схемы RSA, т.е. n является произведением двух простых и e взаимно просто с $\varphi(n)$. Число C является результатом шифрования некоторого сообщения M.

Формат выходных данных

Выведите одно число M ($0 \leqslant M < n$), которое было зашифровано такой криптосхемой.

Примеры

rsa.in	rsa.out
143 113 41	123
9173503 3 4051753	111111

Задача D. Деление кучки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Рассмотрим такую игру: на столе есть кучки из камней. Изначально кучек две, в первой a камней, во второй b камней. За ход можно взять любую кучку и разбить ее на несколько (больше одной) кучек равного размера. Кто выиграет при оптимальной игре?

Формат входных данных

Входной файл содержит два числа a и b ($1 \leqslant a, b \leqslant 10^9$).

Формат выходных данных

Выведите `First`, если выигрывает первый игрок и `Second`, если второй.

Примеры

стандартный ввод	стандартный вывод
12 13	First
15 21	Second

Задача E. НОД прогрессии

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Даны числа a, b и k. Найдите наибольший общий делитель чисел $a, a+b, a+2b, a+3b, \ldots, a+kb$.

Формат входных данных

Во входном файле даны три числа a, b и k ($1 \leqslant a, b, k \leqslant 10^9$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
4 2 3	2

Задача F. Улиточки

Имя входного файла:	`snails.in`
Имя выходного файла:	`snails.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Две улиточки Маша и Петя сейчас находятся на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). Ввиду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Формат входных данных

В первой строке файла записаны четыре целых числа — n, m, a и h (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика).

В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние).

Ограничения: $2 \leqslant n \leqslant 10^5, 0 \leqslant m \leqslant 10^5, s \neq t$

Формат выходных данных

Если существует решение, то выведите `YES` и на двух отдельных строчках сначала путь для Машеньки (т.к. дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите `NO`. Если решений несколько, выведите любое.

Примеры

snails.in	snails.out
3 3 1 3 1 2 1 3 2 3	YES 1 3 1 2 3

Задача G. Проверьте декомпозицию

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам дано неориентированное дерево и его центроидная декомпозиция. Проверьте, что центроидная декомпозиция построена корректно.

Формат входных данных

В первой строке дано число n ($1 \leqslant n \leqslant 200000$) — количество вершин в дереве.

В следующих $n-1$ строках даны пары чисел a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) — рёбра дерева.

В следующих $n-1$ строках даны пары чисел a_i, b_i ($1 \leqslant a_i, b_i \leqslant n$) — ориентированные рёбра центроидной декомпозиции. Гарантируется, что данная декомпозиция является «ориентированным деревом», т. е. подвешенным деревом, в котором рёбра ориентированы от предка к потомку.

Формат выходных данных

Выведите «YES» (без кавычек), если декомпозиция является корректной (т. е. каждая вершина — центроид в соответствующей части дерева), и «NO» (без кавычек) иначе.

Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3 2 1 2 3	YES
3 1 2 2 3 3 1 1 2	NO

Задача H. Стреляем в НЛО

Имя входного файла:	`ufo.in`
Имя выходного файла:	`ufo.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На землю напали инопланетяне! Над землей зависли n летающих тарелок. Будем представлять каждую тарелку отрезком, параллельным оси x, координаты концов i-го отрезка (l_i, h_i) и (r_i, h_i). У нас есть лазерная пушка. Пушка стреляет вертикально вверх (параллельно оси y) и поражает все НЛО, которые пересек ее луч (если луч пересек самый конец отрезка, тоже считается, что НЛО сбито).

Было произведено m выстрелов из пушки, i-й выстрел произведен из точки $(x_i, 0)$. Определите для каждого выстрела, какие НЛО были им сбиты (после того, как НЛО сбит, следующие выстрелы его уже не сбивают).

Формат входных данных

В первой строке задано целое n ($1 \leqslant n \leqslant 200\,000$) — число НЛО. В следующих n строках заданы по три целых числа l_i и r_i и h_i ($-10^9 \leqslant l_i \leqslant r_i \leqslant 10^9$, $1 \leqslant h_i \leqslant 10^9$).

В следующей строке записано число выстрелов m ($1 \leqslant m \leqslant 200\,000$). Далее, в m строках заданы координаты x_i ($-10^9 \leqslant x_i \leqslant 10^9$).

Формат выходных данных

Для каждого выстрела в своей строке выведите c_i — число НЛО, сбитых выстрелом x_i, а затем c_i чисел $a_{i1} < a_{i2} < \ldots < a_{ic_i}$ — номера этих НЛО.

Примеры

ufo.in	ufo.out
4 10 30 10 30 50 50 40 60 30 70 80 40 2 50 30	2 2 3 1 1

Задача I. Генерируем отрезки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сгенерируйте n попарно непересекающихся и непараллельных отрезков на плоскости.

Формат входных данных

В первой строке дано целое число n ($1 \leqslant n \leqslant 100000$) — количество отрезков.

Формат выходных данных

В n строках выведите по четыре целых числа x_1, y_1, x_2, y_2 — координаты концов отрезков. Координаты должны не превосходить 10^6 по абсолютной величине.

Примеры

стандартный ввод	стандартный вывод
3	1 1 2 2 3 2 4 1 2 0 3 0

Задача J. Ним в поддавки

Имя входного файла:	`invnim.in`
Имя выходного файла:	`invnim.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Всем вам хорошо известна игра ним: на столе лежит кучка из a камней, своим ходом можно взять из неё любое число камней от 1 до a. Однако, эта игра необычная: если обычно игрок, который не может сделать ход, проигрывает, здесь всё наоборот — игрок, который не может сделать ход.

Как и в привычном вам варианте игры, на столе лежит не одна кучка камней, а n. Каждый игрок своим ходом может взять из любой кучки любое число камней. Вам нужно определить, кто победит в этой игре.

Формат входных данных

В первой строке задано целое число n ($1 \leqslant n \leqslant 100$). Во второй строке заданы размеры кучек a_i ($1 \leqslant a_i \leqslant 1000$).

Формат выходных данных

В единственной строке выведите «`FIRST`», если победит первый игрок, или «`SECOND`», если победит второй.

Примеры

invnim.in	invnim.out
2 1 2	FIRST
1 1	SECOND

Задача K. Кредитные операции

Имя входного файла:	`credit.in`
Имя выходного файла:	`credit.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Крупный предприниматель Владимир Дубинин, в недалёком прошлом больше известный как Вован Палёный, контролирует трест из N предприятий. Бывший подельник Владимира, а ныне известный банкир Александр Кулаков по прозвищу Саня Кривой владеет холдингом из N банков. Как и полагается у старых друзей, предприятия г-на Дубинина берут кредиты исключительно у банков г-на Кулакова, в то время как банки г-на Кулакова выдают кредиты только предприятиям г-на Дубинина. Причём с целью уклонения от уплаты налогов вся информация о размерах кредитов тщательно скрывается. Но тут на сцене появляется давний конкурент Владимира и Александра генерал милиции Иван Ломов, когда-то носивший кличку Ваня Гнилой. Г-н Ломов хочет отомстить г-ну Дубинину и г-ну Кулакову за старые обиды и выявить все кредитные операции между их предприятиями и банками.

Для начала люди Ивана произвели выемку документов из офисов предприятий Владимира и для каждого предприятия выяснили суммарный размер кредитов $SR[i]$, полученных этим предприятием. Затем в ходе аналогичных операций для каждого банка Александра был установлен суммарный размер кредитов $SC[j]$, выданных этим банком. Последним шагом является заполнение так называемой кредитной матрицы на основе полученных данных. В данном случае кредитная матрица представляет собой квадратную таблицу из N строк и N столбцов, в которой каждый элемент $A[i, j]$ должен быть равен размеру кредита, взятого i-м предприятием г-на Дубинина у j-го банка г-на Кулакова. Доподлинно известно, что размер любого кредита является целым числом от 0 до 100. Помните, что полученная в ходе следственных действий информация могла быть сфальсифицирована, и тогда заполнить кредитную матрицу не получится.

Формат входных данных

Первая строка содержит целое число N ($2 \leqslant N \leqslant 100$). Вторая строка содержит N целых чисел $SR[i]$ ($0 \leqslant SR[i] \leqslant 32000$). Третья строка содержит N целых чисел $SC[j]$ ($0 \leqslant SC[j] \leqslant 32000$). Сумма всех $SR[i]$ равняется сумме всех $SC[j]$.

Формат выходных данных

Выведите «`NO`», если кредитная матрица не может быть заполнена. Иначе в первой строке выведите «`YES`», а в каждой из следующих N строк выведите через пробел N соответствующих элементов $A[i, j]$ кредитной матрицы. Если задача имеет несколько решений, можно вывести любое из них.

Примеры

credit.in	credit.out
4 267 157 188 259 193 320 346 12	YES 100 55 100 12 0 70 87 0 0 95 93 0 93 100 66 0